

IDENTIFICATION AND PROACTIVE PREVENTION OF PHISHING WEBSITE

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

BY

R.SRINATH-38110557

AND

A.V.SUDARSAN-38110566



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

MARCH 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **R.SRINATH (Reg. No:38110557)** and **A.V.SUDARSAN (Reg.No:38110556)** who carried out the project entitled **"IDENTIFICATION AND PROACTIVE PREVENTION OF PHISHING WEBSITE"** under my supervision from November 2021 to March 2022.

Internal Guide
Dr. M. KANIPRIYA PhD

Head of the Department
Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **R.SRINATH** and **A.V.SUDARSAN** hereby declare that the project report entitled **“IDENTIFICATION & PROACTIVE PREVENTION OF PHISHING WEBSITE”** done by us under the guidance of **Dr. M.KANIPRIYA** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. M.KANIPRIYA, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways .

ABSTRACT

Phishing is a fraudulent process or an attempt to steal one's personal information. Phishing usually occurs via email or by portraying website as a legitimate one. In order to stop the phishing attempt, the solution is to either find or recognize the phish. The solution to the problem requires Random Forest (RF), one of the different types of machine learning based algorithms used for detection of Phishing websites. The performance of the regressor are compared and measured in terms of accuracy and with the help of values generated from given conditions are predicted. The proposed model is provided with an accuracy of 79% and a combination of 17 features.

Keywords— Phishing website detection, Random Forest Regressor, Train test Split, Machine Learning, Cross Validation.

LIST OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	1
	LIST OF FIGURES	2
1	INTRODUCTION	
1.1	GENERAL	3
1.2	OBJECTIVE	4
1.3	EXISTING SYSTEM	5
1.3.1	DRAWBACKS IN EXISTING SYSTEM	5
1.4	PROPOSED SYSTEM	6
1.4.1	ADVANTAGES OF PROPOSED SYSTEM	6
2	LITERATURE SURVEY	6
3	SYSTEM DESIGN	9
3.1	GENERAL	9
3.1.1	SYSTEM ARCHITECTURE DIAGRAM	9
3.1.2	USE CASE DIAGRAM	10
3.1.3	CLASS DIAGRAM	11
3.1.4	ACTIVITY DIAGRAM	12
3.1.5	SEQUENCE DIAGRAM	13
4	PROJECT DESCRIPTION	13
4.1	GENERAL	13
4.2	MODULE DESCRIPTION	14
4.2.1	SIGN IN MODULE	14
4.2.2	REGISTER MODULE	14

4.2.3	PREDICTION MODULE	15
4.2.3.1	DATASET AND FEATURES	15
4.2.3.2	WORKING PROCEDURE	22
4.3	DEVELOPMENT ENVIRONMENT	24
4.3.1	HARDWARE REQUIREMENTS	24
4.3.2	SOFTWARE REQUIREMENTS	25
5	CONCLUSION AND FUTURE ENHANCEMENT	25
5.1	CONCLUSION	25
5.2	FUTURE ENHANCEMENT	25
6	REFERENCES	26
7	APPENDIX	26
7.1	SCREENSHOTS	26
7.2	SOURCE CODE	34

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Phishing is a type of extensive fraud that happens when a malicious website act like a real one keeping in mind that the end goal to obtain touchy data, for example, passwords, account points of interest, or MasterCard numbers.

In spite of the fact that there are a few contrary to phishing programming and methods for distinguishing potential phishing endeavours in messages and identifying phishing substance on sites, phishes think of new and half breed strategies to go around the accessible programming and systems.

Phishing is a trickery system that uses a blend of social designing what's more, innovation to assemble delicate and individual data, for example, passwords and charge card subtle elements by taking on the appearance of a dependable individual or business in an electronic correspondence. Phishing makes utilization of spoof messages that are made to look valid and implied to be originating from honest to goodness sources like money related foundations, ecommerce destinations and so forth, to draw clients to visit fake sites through joins gave in the phishing email. The misleading sites are intended to emulate the look of a genuine organization site page.

The employing so as to phishing invader's trap clients diverse social building strategies, for example, debilitating to suspend client accounts on the off chance that they don't finish the account upgrade process, give other data to approve their records or a few different motivations to get the clients to visit their satirize page.

Supervised learning (Classification Technique) accommodates a vastly improved precision while unsupervised learning accommodates a quick and dependable way to deal with infer information from a dataset. That's why we used supervised learning in our work.

1.2 SCOPE AND OBJECTIVE

The system should be useful in many e-commercial websites for maintaining the security and reliability

of customers and people online

The system should be useful in preventing online frauds leading to leakage of important and private user data

The scope of using Machine Language over other Traditional Detecting Methods

Objectives:

- Understanding phishing domain (or Fraudulent Domain) characteristics, its distinguishing features from legitimate domains
- Why it is so important to detect this domain and how they can be detected using machine learning and natural language processing techniques
- Reviewing the state-of-the-art machine learning techniques for malicious URL detection in literature
- Understanding the newly emerging concept of Malicious URL Detection as a service and the principles to be used while designing such a system.

To distinguish the phishing websites from the legitimate websites and ensure secure transactions to users.

1.3 EXISTING SYSTEM

In existing system there are two different ways to detect a phishing website, they are by manually checking and other one is algorithm based detection. By manual checking of url link implies if we know the webpage which is considered as the target webpage, we can confirm which all are the phishing pages and with the help of comments for the website it is possible to detect the website is genuine or not. Another method is by using algorithms like Random Forest Classifier, Data Mining etc to detect Phishing Website.

1.3.1 DRAWBACKS IN EXISTING SYSTEM

- Manual checking takes more time
- Hard to find original webpage by manual checking.

- Takes more time to process.

1.4 PROPOSED SYSTEM

The proposed System mainly uses three features.They are train test split,Random Forest Regressor and Cross Validation.Phishing websites can be detected easily with the help of these features.The website given is classified into 17 conditions and we perform the above mentioned three features on the dataset present to train the model and get accuracy of the model.Once the model is trained then the 17 conditions output for a website is provided to the model to get the output.

1.4.1 ADVANTAGES IN PROPOSED SYSTEM

- Accuracy is improved.
- Time consumption is less.
- Suspicious websites are also found.

CHAPTER 2

LITERATURE SURVEY

2.1 Detecting Phishing Websites Using Machine Learning - Amani

Alswailem,Bashayr Alabdullah and Norah Alrumayh

This paper proposes random forest technique to detect phishing website.The combination of 36 features are used to train the model and to generate value for the given URL.The 36 features can be categorized into three main categories.They are features from URL,features from page content and features from page rank.The URL and page rank features can be extracted from URL object where page content features can be extracted from DOM object.By splitting the datasets into train data and test data they classify the given URL with the help of trained data model.

2.2 Detection of Phishing URLs Using Machine Learning Techniques - Joby James ,Ciza Thomas and Sandhya L

This paper proposed that domain names and phishing URL have different length compared to legitimate URL. It is used to detect phishing websites based on Naive Bayes's classifier or SVM or Regression classifier or KNN. The host based, page based and lexical page based property were applied to the URL to form a featured valued datasets. The feature value for a phishing website is 1 and feature value for a benign website is 0. The dataset is splitted into train and test files. The classifier is decided by the WEKA and using MATLAB. By using any one of the classification algorithm it can classify the given input URL as a phishing or not based on the trained model.

2.3 PhishMon: A Machine Learning Framework for Detecting Phishing Web pages-Amirreza Niakanlahiji, Bei-Tseng Chu and Ehab Al-Shaer

In this paper, 18 features are used for data preprocessing. Out of which 15 of them are new features. The 18 features can be categorized into three main categories. They are HTTP features, code complexity features and certificate features. Once the given URL is preprocessed, the 18 feature based values are generated. It uses nearly four machine learning algorithms for classification. They are CART, K-Nearest Neighbours, AdaBoost and Random Forest. The area under the ROC curve is used as a performance measure to compare the classifier. It also uses 10 fold-cross validation to estimate the performance of the classifiers. Once the dataset is trained to these four model, the 18 feature based value for input URL is given to all four classifiers. Based on the input values it classifies the given input URL with the accuracy of 95.4%.

2.4 Phishing Detection from Urls by Using Neural Networks - Ozgur Koray Sahingoz, Saide Işıl原因 Baykal and Deniz Bulut

The author proposed this paper used 37,175 phishing and 36,400 legitimate

websites and the model is trained using Artificial Neural Network(ANN) and Deep Neural Network(DNN).The websites are classified with the help of 16 features combination.In case of ANN the dataset is trained with the help of RELU and TANH activation function with one hidden layer and 20 neurons.In case of DNN the activation function is RELU and 40 neurons and it provides 91% accuracy.Once the training is done,the given URL is cross validated and the dataset is splitted into training and test data.Then the URL is classified using SVM to predict the output.

2.5 Detection and Prevention of Phishing websites using Machine Learning -

Vaibhav Patil Pritesh Thakkar Pritesh Thakkar Tushar Bhat

Phishing websites are detected using three machine learning algorithm in this paper.First the URL is compared using Black list and white list approach.If the URL is found then the result is published.Then the features are extracted using Heuristic approach or visual similarity approach.Once the features are selected, the values based on these features for the input URL is also generated.Then the dataset is trained to Random Forest and Decision tree models.The feature based generated values for input URL is given to the logistic regression and other two models and it predicts the output with greater accuracy.

2.6 A Survey of URL-based Phishing Detection - Eint Santi Aung, Chaw Thet zan and Hayato Yamana

Phishing websites are detected using Convolutional Neural Network(CNN).In this paper,they followed nearly five approaches.They are White list based approach,black list based approach,visual similarity based approach,content based approach and URL based approach.The features extracted are based on URL property only.The feature based value for input is generated and datasets are given to four models for training.The four machine learning algorithm used are Naive Bayes's Support Vector Machine(SVM),Random Forest and Convolutional Neural Network(CNN).

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

System Design deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

3.1.1 SYSTEM ARCHITECTURE DIAGRAM

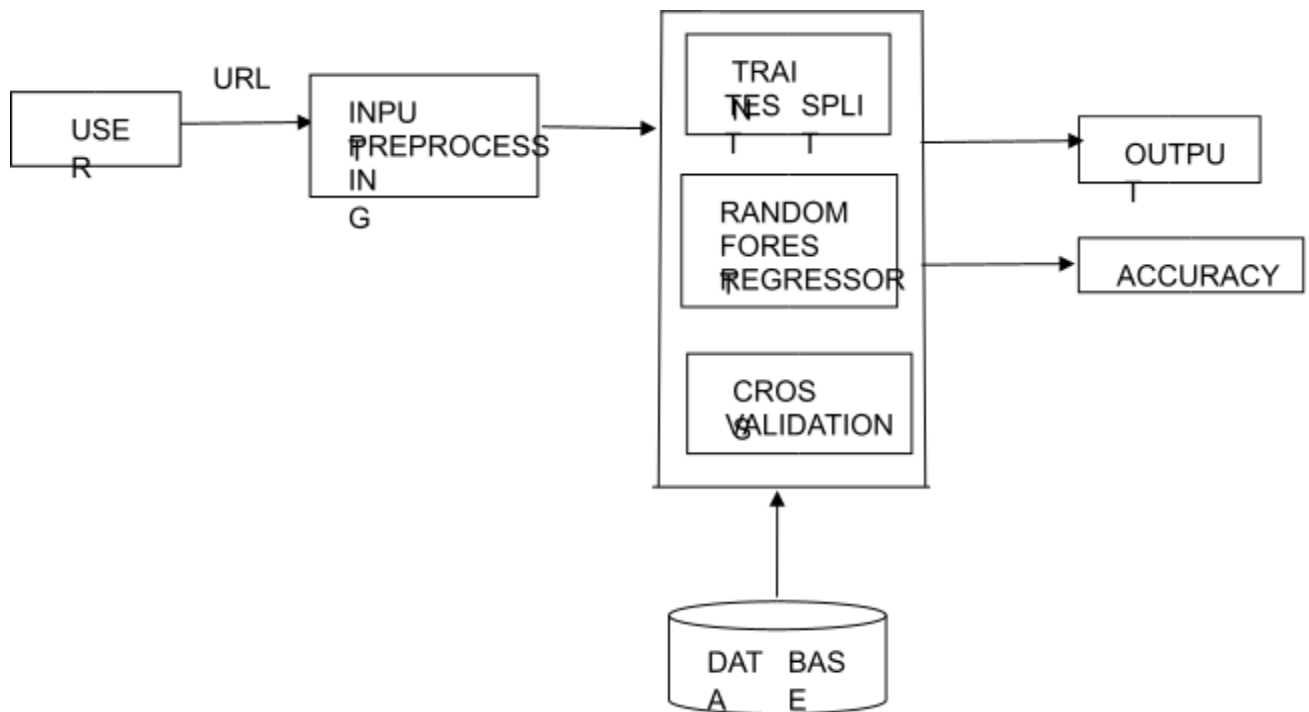


Figure 3.1.1 System Architecture Diagram for Phishing website detection

3.1.2 USE CASE DIAGRAM

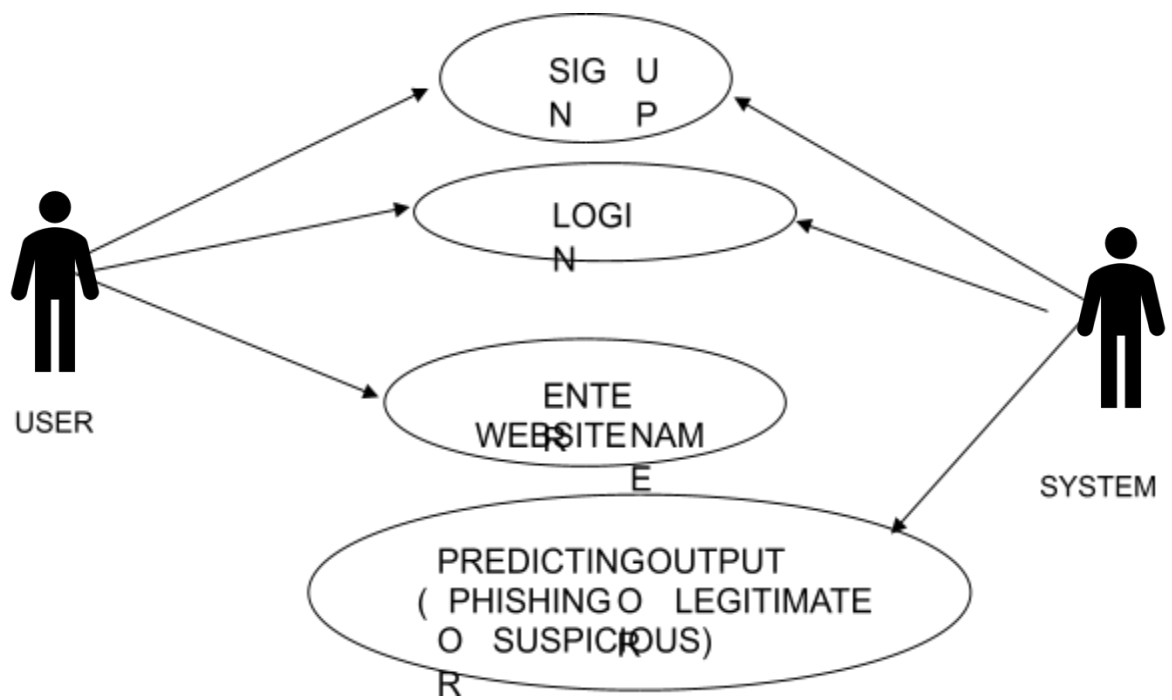


Figure 3.1.2 Use Case Diagram for Phishing website detection

3.1.3 CLASS DIAGRAM

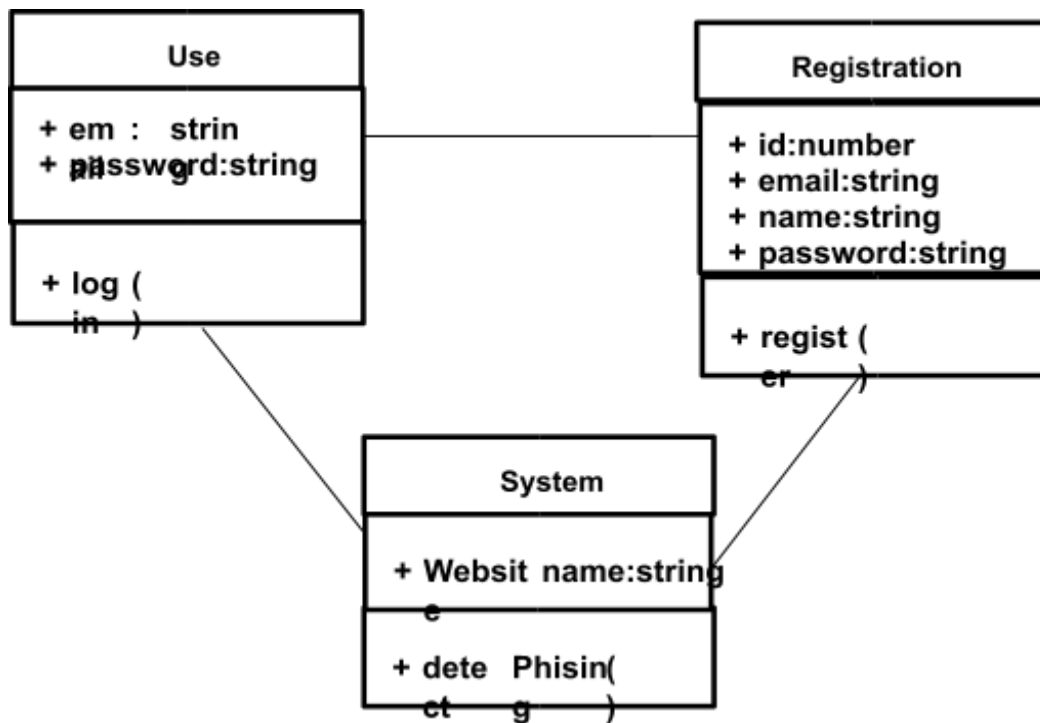


Figure 3.1.3 Class Diagram for Phishing Website detection

3.1.4 ACTIVITY DIAGRAM

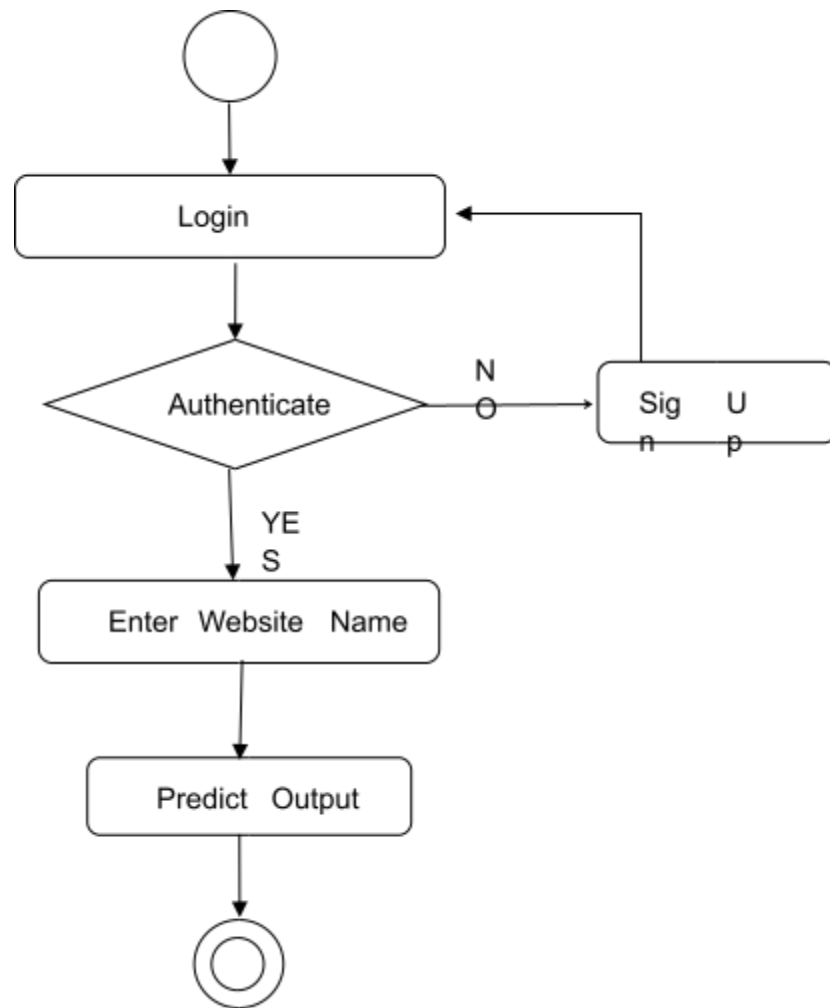


Figure 3.1.4 Activity Diagram for Phishing Websites detection

3.1.5 SEQUENCE DIAGRAM

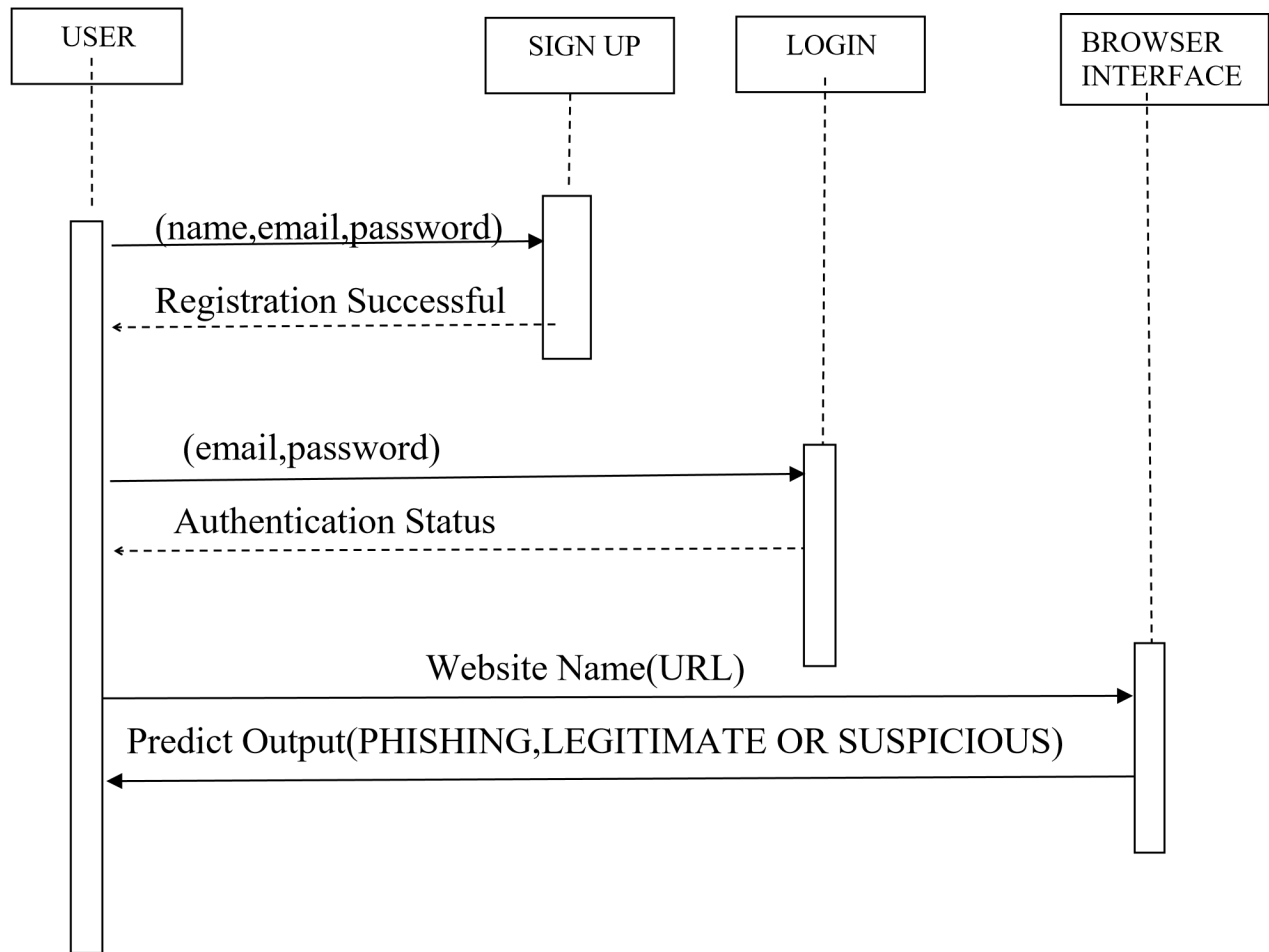


Figure 3.1.5 Sequence Diagram for Phishing Websites detection

CHAPTER 4

PROJECT DESCRIPTION

4.1 GENERAL

Detection of Phishing Websites. This is an application used to predict whether a website is a phishing website or not. This application is created by using Python. Django framework is used for developing the back end of the application. User interface is created by using bootstrap application. This application uses a dataset. We perform three operations on the dataset for

predicting the Output. They are Train Test Split, Random Forest Regression and Cross Validation. After these methods are performed sequentially, then the value for the given url from the 17 conditions is given to the model to get the output on whether the given website is a phishing website or not.

4.2 MODULE DESCRIPTION

Firstly the application will ask to login in order to use it. If already an user, use your username and password to login. If you are a new user, create an user ID by registering. After logging into the application, a textbox will appear where we can enter the link of the website we want to check. After entering the link of the website we hit check. After that a new window will open. The window will display whether the given website is a phishing website or not.

4.2.1 SIGN IN MODULE

The sign in module is for the user to sign in to use the application. The sign in module is the typical sign in module used for every other applications and websites. The username and password will already be stored in the database. If the username and password matches, access will be granted.

4.2.2 REGISTER MODULE

The register module is used to create a new user. In order to create a new user, following details are needed.

1. Desired Username
2. E-Mail ID
3. A Password

After all the details are filled out. Hit register. The details are stored in the database. Hence a new user is created.

4.2.3 PREDICTING MODULE

This is the text box that appears after logging into the application. We insert the link of the website we want to check in the text box and hit check. A new window will open with the result.

The predicting module uses a dataset. And the dataset is trained according to a set of conditions which would help in detecting a phishing website.

4.2.3.1 DATASET AND FEATURES

A data set is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

One of the challenges faced by our research was the unavailability of reliable training datasets. In fact, this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites using data mining techniques have been disseminated these days, no reliable training dataset has been published publically, maybe because there is no agreement in literature on the definitive features that characterize phishing websites, hence it is difficult to shape a dataset that covers all possible features. In this article, we shed light on the important features that have proved to be sound and effective in predicting phishing websites. In addition, we proposed some new features, experimentally assign new rules to some well-known features and update some other features.

Those features are :

1. Address bar -Based Features
2. Domain-Based Features

3. Abnormal-Based Features

I. Address Bar-Based Features:

1. Using the IP Address:

If an IP address is used as an alternative of the domain name in the URL, such as "<http://125.98.3.123/fake.html>", users can be sure that someone is trying to steal their personal information. Sometimes, the IP address is even transformed into hexadecimal code as shown in the following link "<http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html>".

Rule: IF { If The Domain Part has an IP Address → Phishing
Otherwise → Legitimate

2. Long URL to Hide the Suspicious Part:

Phishers can use long URL to hide the doubtful part in the address bar. For example:

http://federmacedoadv.com.br/3f/aze/ab51e2e319e51502f416dbe46b773a5e/?cmd=_home&dispatch=11004d58f5b74f8dc1e7c2e8dd4105e811004d58f5b74f8dc1e7c2e8dd4105e8@phishing.website.html

To ensure accuracy of our study, we calculated the length of URLs in the dataset and produced an average URL length. The results showed that if the length of the URL is greater than or equal 54 characters then the URL classified as phishing. By reviewing our dataset we were able to find 1220 URLs lengths equals to 54 or more which constitute 48.8% of the total dataset size.

Rule: IF { URL length < 54 → feature = Legitimate
Else if { URL length >= 54 and ≤ 75 → feature = Suspicious
Otherwise → feature = Phishing

We have been able to update this feature rule by using a method based on frequency and thus improving upon its accuracy.

3.Using URL Shortening Services “TinyURL”:

URL shortening is a method on the “World Wide Web” in which a URL may be made considerably smaller in length and still lead to the required webpage. This is accomplished by means of an “HTTP Redirect” on a domain name that is short, which links to the webpage that has a long URL. For example, the URL

“http://portal.hud.ac.uk/” can be shortened to “bit.ly/19DXSk4”.

Rule:IF { TinyURL → Phishing
Otherwise → Legitimate

4.URL’s having “@” Symbol:

Using “@” symbol in the URL leads the browser to ignore everything preceding the “@” symbol and the real address often follows the “@” symbol.

R
u
l
e
:
I
f

{ Url Having @ Symbol → Phishing

Otherwise → Legitimate

5.Redirecting using “//”:

The existence of “//” within the URL path means that the user will be redirected to another website. An example of such URL’s is:

“http://www.legitimate.com//http://www.phishing.com”. We examine the location where the “//” appears. We find that if the URL starts with “HTTP”, that means the “//” should appear in the sixth position. However, if the URL employs “HTTPS” then the “//” should appear in seventh position.

Rule: IF The Position of the Last Occurrence of "/" in the URL $\geq 7 \rightarrow$
 Phishing Otherwise $\rightarrow$ Legitimate

6. Adding Prefix or Suffix Separated by (-) to the Domain:

The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example

<http://www.Confirme-paypal.com/>.

{ DomainNamePartIncludes(-)Symbol $\rightarrow$ Phishing
 Rule: IF Otherwise $\rightarrow$ Legitimate

7. Domain Registration Length:

Based on the fact that a phishing website lives for a short period of time, we believe that trustworthy domains are regularly paid for several years in advance. In our dataset, we find that the longest fraudulent domains have been used for one year only.

years { Domains Expires on ≤ 1
 $\rightarrow$ Phishing
 Rule: IF Otherwise $\rightarrow$
 Legitimate

8. Using Non-Standard Port:

This feature is useful in validating if a particular service (e.g. HTTP) is up or down on a specific server. In the aim of controlling intrusions, it is much better to merely open ports that you need. Several firewalls, Proxy and Network Address Translation (NAT) servers will, by default, block all or most of the ports and only open the ones selected. If all ports are open, phishers can run almost any service they want and as a result, user information is threatened. The most important ports and their preferred status are shown in Table 1.

{ Port#isofthePreferredStatus $\rightarrow$ Phishing
 Rule: IF Otherwise $\rightarrow$ Legitimate

PORT	SERVICE	MEANING	PREFERRED STATUS
21	FTP	Transfer files from one host to another	Close
22	SSH	Secure File Transfer Protocol	Close

23	Telnet	provide a bidirectional interactive text-oriented communication	Close
80	HTTP	Hyper text transfer protocol	Open
443	HTTPS	Hypertext transfer protocol secured	Open
445	SMB	Providing shared access to files, printers, serial ports	Close
1433	MSSQL	Store and retrieve data as requested by other software applications	Close
1521	ORACLE	Access oracle database from web.	Close
3306	MySQL	Access MySQL database from web.	Close
3389	Remote Desktop	allow remote access and remote collaboration	Close

Table 1 Common ports to be checked

9.The Existence of “HTTPS” Token in the Domain Part of the URL:

The phishers may add the “HTTPS” token to the domain part of a URL in order to trick users. For example,

<http://https-www-paypal-it-webapps-mpp-home.soft-hair.com/>.

→ { Using HTTP Token in Domain Part of The URL
Rule: IF Phishing Otherwise → Legitimate

II. Domain based Features:

10.Age of Domain:

This feature can be extracted from WHOIS database (Whois 2005). Most phishing websites live for a short period of time. By reviewing our dataset, we find that the minimum age of the legitimate domain is 6 months.

6months { AgeOfDomain ≤
→ Legitimate Rule:
IF Otherwise → Phishing

11.DNS Record:

For phishing websites, either the claimed identity is not recognized by the WHOIS database (Whois 2005) or no records founded for the hostname (Pan and Ding 2006). If the DNS record is empty or not found then the website is classified as “Phishing”, otherwise it is classified as “Legitimate”.

No DNS { Record For The Domain →
Phishing Rule: IF Otherwise →
Legitimate

12. Statistical-Reports Based Feature:

Several parties such as PhishTank(PhishTank Stats, 2010-2012), and

StopBadware(StopBadware, 2010-2012) formulate numerous statistical reports on phishing websites at every given period of time; some are monthly and others are quarterly. In our research, we used 2 forms of the top ten statistics from PhishTank: “Top 10 Domains” and “Top 10 IPs” according to statistical-reports published in the last three years, starting in January 2010 to November 2012. Whereas for “StopBadware”, we used “Top 50” IP addresses.

Rule: IF Host Belongs to Top Phishing IPs or Top Phishing Domains → Phishing }
Otherwise → Legitimate

III. Abnormal Based Features:

13. Request URL:

Request URL examines whether the external objects contained within a webpage such as images, videos and sounds are loaded from another domain. In legitimate webpages, the webpage address and most of objects embedded within the webpage are sharing the same domain.

Rule: IF { % of Request URL < 22% → Legitimate
% of Request URL ≥ 22 and ≤ 61% → Suspicious
Otherwise → Phishing

14. URL of Anchor:

An anchor is an element defined by the <a> tag. This feature is treated exactly as “Request URL”. However, for this feature we examine:

1. If the <a> tags and the website have different domain names. This is similar to request URL feature.
2. If the anchor does not link to any webpage, e.g.:

A.

B.

C.

D.

% of URL
Rule: IF

{	of Anchor < 31% → Legitimate % of URL of Anchor >=31% and <=67% → Suspicious Otherwise → Phishing
---	---

15. Links in <Meta>, <Script> and <Link> tags:

Given that our investigation covers all angles likely to be used in the webpage source code, we find that it is common for legitimate websites to use <Meta> tags to offer metadata about the HTML document; <Script> tags to create a client side script; and <Link> tags to retrieve other web resources. It is expected that these tags are linked to the same domain of the webpage.

Rule: IF%

{	of Links in "< Meta>","<script>" and "<Link>" < 17% → Legitimate % of Links in "< Meta>","<script>" and "<Link>" >=17%→ Suspicious and <=81% Otherwise → Phishing	}
---	---	---

16. Server Form Handler (SFH):

SFHs that contain an empty string or "about:blank" are considered doubtful because an action should be taken upon the submitted information. In addition, if the domain name in SFHs is different from the domain name of

the webpage, this reveals that the webpage is suspicious because the submitted information is rarely handled by external domains.

$$\left\{ \begin{array}{l} \text{SFH is "about: blank" Or Is Empty} \rightarrow \text{Phishing} \\ \text{Rule: IF } \text{SFHRefersToADifferentDomain} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{array} \right.$$

17.Abnormal URL:

This feature can be extracted from WHOIS database. For a legitimate website, identity is typically part of its URL.

$$\text{Rule: } \left\{ \begin{array}{l} \text{The Host Name Is Not Included In URL} \rightarrow \text{Phishing} \\ \text{IF Otherwise} \rightarrow \text{Legitimate} \end{array} \right.$$

4.2.3.2 WORKING PROCEDURE

Dataset is generally classified on 17 features. Number of dataset for our project is 11,055. Number of phishing website in our dataset is 6,812 and number of legitimate website in our dataset is 4,243.

A.Splitting dataset using Train Test split:

The first task after preprocessing is to split dataset into train and test data..The dataset is splitted with the help of train test split from sklearn package.The dataset contains 6,157 legitimate website and 4,898 phishing website.In our case we provide the test_size parameter as 0.2.So it splits the 11,056 data into 8,845 training data (i.e 80%) and 2,211 test data(i.e 20%) .

B.Training and testing using dataRandom Forset Regressor:

Once the dataset is splitted using traintest split the train data is provided to Random Forest Regressor(RFR) and the model is trained.The Random Forest Regressor (RFR) is imported from sklearn package.The model is trained with the help of decision trees.The training data is splitted and provided to each decision tree for training.In our case,we provided the number of decision tree as 10.So each 10 decision tree is trained of different

training data inputs. Once the model is trained we provide the test data to the model. So the test data is splitted and provided to 10 decision trees. Each decision tree provides a continuous value for the test data based on the model trained. The continuous value from each decision tree is taken and average is calculated. From the average it is predicted that it provides correct value for a test data or not.

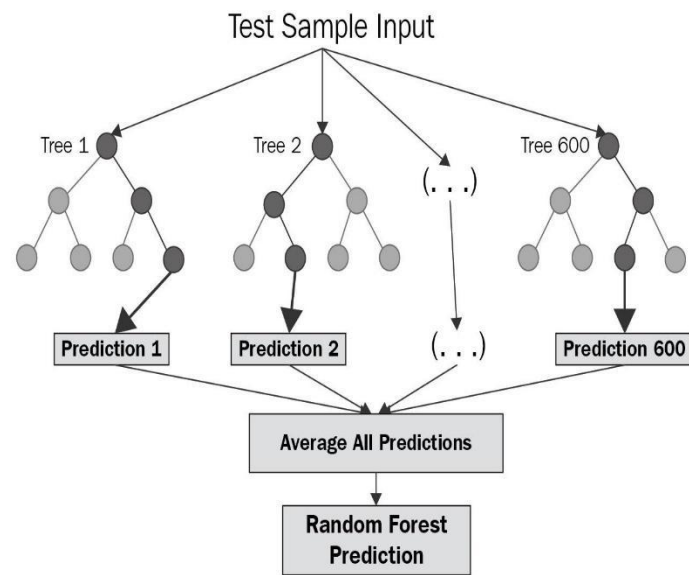


Fig 4.2.3.1.1 Random Forest Regressor

C. Check Accuracy using Cross Validation:

After training the model, the dataset is cross validated to check the accuracy. The cross validation is imported from sklearn package. The dataset is splitted based on the k-fold value specified in cross validation. In our case, we provide the k-fold value is 5. So the dataset is splitted into 5 parts. Once the dataset is splitted into five parts. Any four of the five part is taken and it is used to train the model. The remaining one part is used as test data. Once the model is trained the test data is used to predict the value. By taking anyone of the four part in previous train data as test data and other three as train data. This process is continued untill all part is treated as test data. Each value observed during the process is used to calculate the

accuracy.

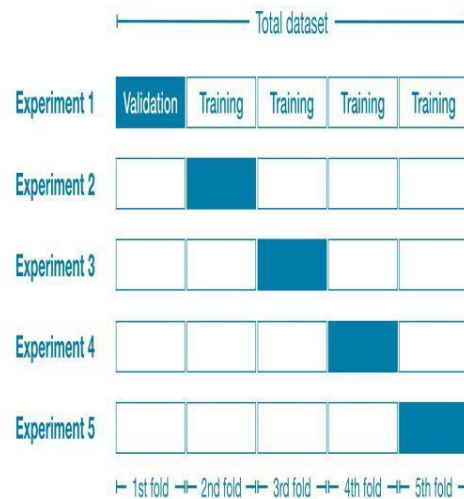


Fig 4.2.3.2 Cross Validation

D.Predicting Output:

The values generated for the given URL from the 17 features is taken as test data and provided to the model to predict the output. Each decision tree in the model takes the test data which will provide some decimal value. The value from each decision tree is taken and averaged to get some final value.

- i) If the average value from the model lies between in the range of -1 to -0.1 or less than -1 then it is classified as Phishing Website.
- ii) If the average value from the model lies between in the range of 0 to 0.9 then it is called as Suspicious Website.
- iii) If the average value from the model is greater than or equal to 1 then it is a Legitimate Website.

4.4 DEVELOPMENT ENVIRONMENT

4.4.1 HARDWARE SPECIFICATION

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software

engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

Processor	Pentium IV Or Higher
Memory Size	500 GB (Minimum)
RAM	8 GB (Minimum)

4.4.2 SOFTWARE SPECIFICATION

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team and tracking the team's progress throughout the development activity.

Operating system	Windows 10
Tool	Python 3.7, Train test split, Random Forest Regressor, Cross Validation

5.CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

Phishing is a cyber crime procedure utilizing both social building and specialized deception to take individual sensitive data. Experimentations against recent dependable phishing data sets utilizing different classification algorithm have been performed which received different learning methods. The aim of this project is to predict whether a given URL is a phishing website or not and regressor (Random Forest Regressor) can be used to produce output with prediction accuracy of 79% by using the given dataset.

5.2 FUTURE ENHANCEMENT

- Every application always needs enhancement to meet the user requirement changes.
- As a future work we might use this model to other Phishing dataset with larger size, now and then testing the performance of those regression algorithm's in terms of prediction accuracy.
- The features that has less impact on result can be changed using feature reduction algorithm.

- New features that can increase the accuracy of predicting the output and reduces the computation time can be added. The new feature is beyond those 36 features.

6. References

1. Amani Alswailem,,Bashayr Alabdullah and Norah Alrumayh. "Detecting Phishing Websites Using Machine Learning"[2019]
2. James Joby, Sandhiya.L, Ciza Thomas. "Detection of Phishing URLs Using Machine Learning Techniques" [2013]
3. Amirreza Niakanlahiji, Bei-Tseng Chu, Ehab Al-Shaer. "PhishMon: A Machine Learning Framework for Detecting Phishing Webpages" [2018]
4. Ozgur Koray Sahingoz, Saide Işıl原因 Baykal and Deniz Bulut " Phishing Detection from Urls by Using Neural Networks"[2018]
5. Vaibhav Patil Pritesh Thakkar Pritesh Thakkar Tushar Bhat. "Detection and Prevention of Phishing Websites using Machine Learning Approach"[2018]
6. Eint Sandi Aung, Chaw Thet Zan, Hayato YAMANA. " A Survey of URL- based Phishing Detection" [2019]

7.APPENDIX

7.1 SCREENSHOTS

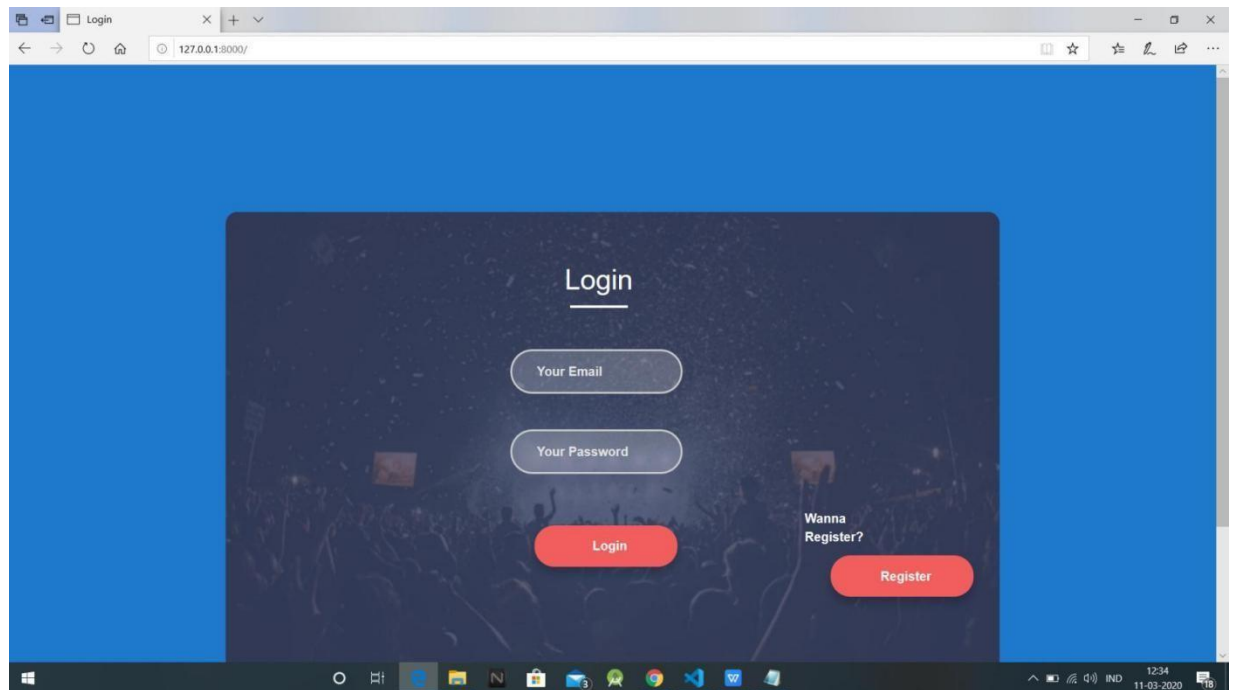


Fig 5.1.1 Login Page

The login page is for the user to sign in to use the application. If the credentials are matched it directs to the main page. Else the error message is displayed. By clicking register button user can move to signup page.

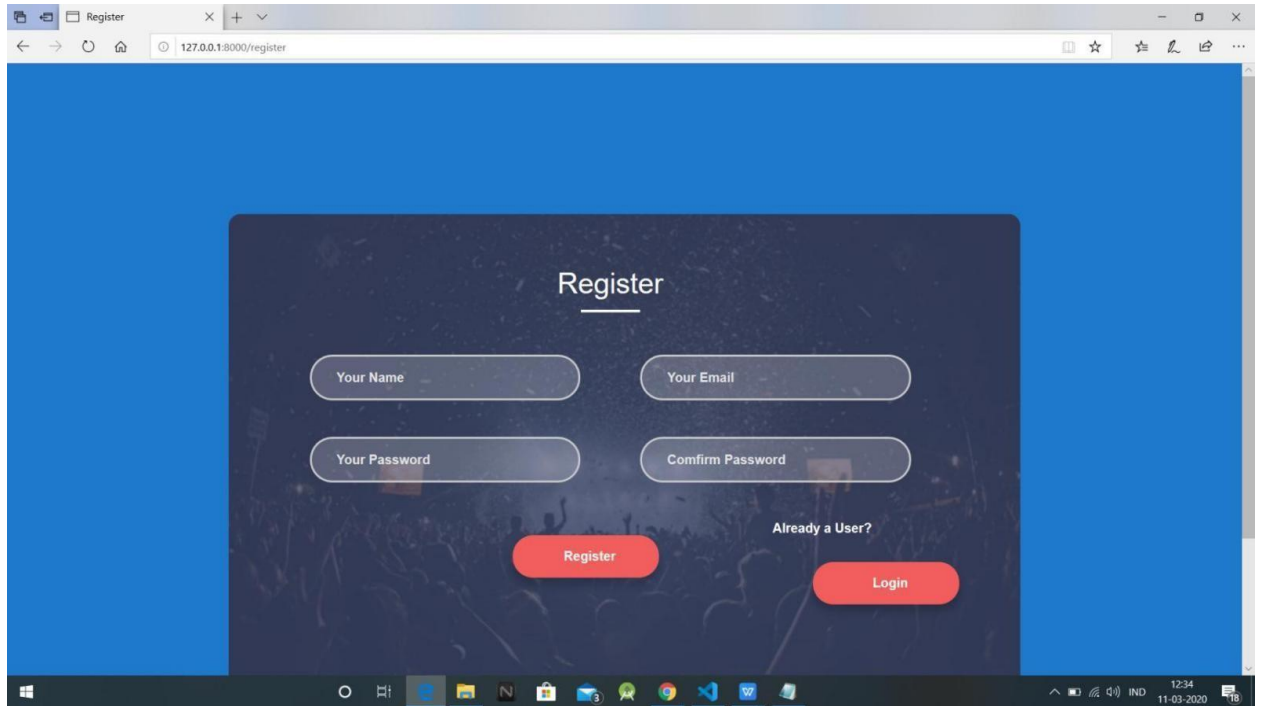


Fig 5.1.2 SignUp Page

The signup page is used to create a new user. If the password and confirm password gets matched it stores the details of the user and it redirects to the login page. Else the error message is displayed.

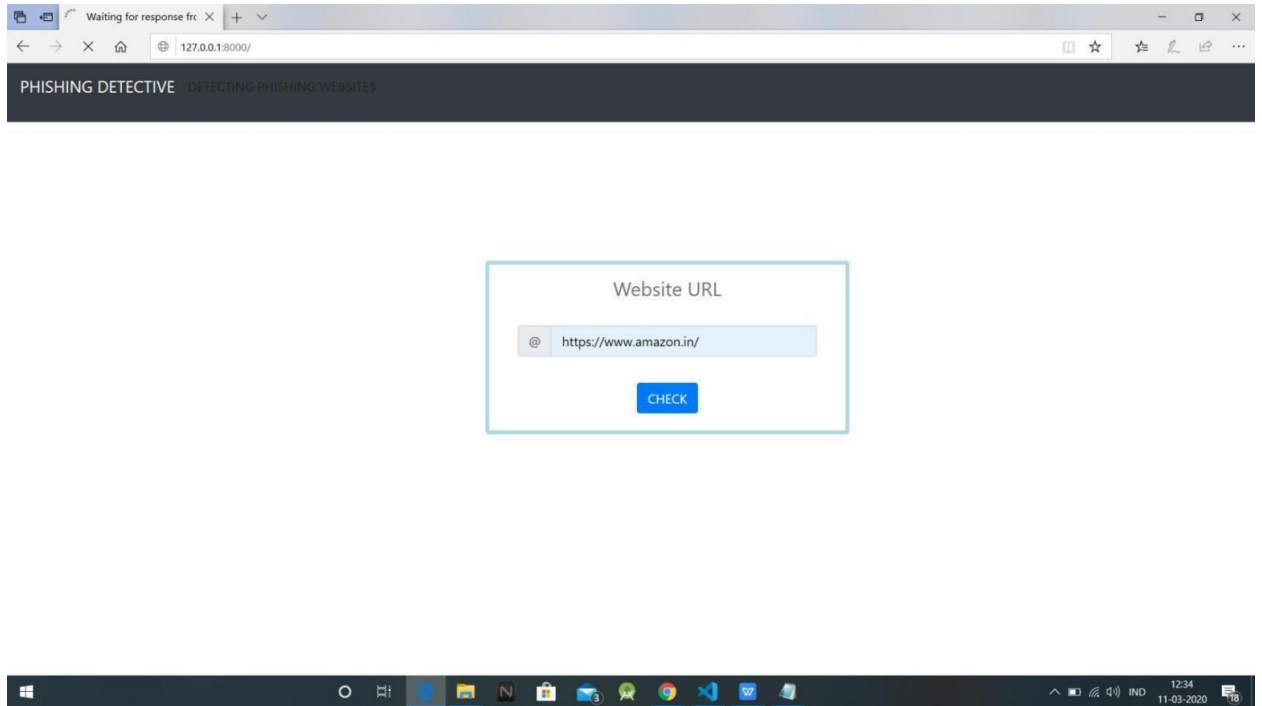


Fig 5.1.3 Index Page

The index page enables the user to provide the url which the user wants to check. Once the check button is pressed it performs all process in the background and produce output.

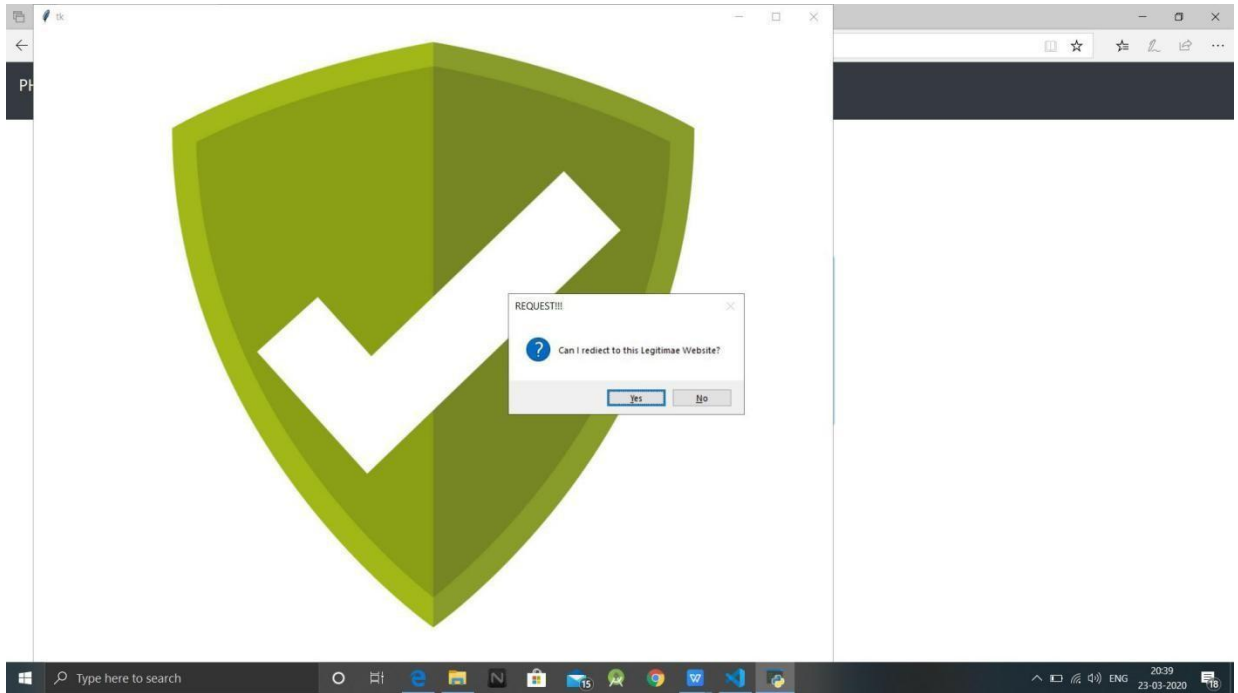


Fig 5.1.4 Legitimate website Output

This is the output page displayed if the given input url is a legitimate one. It also asks the user to redirect to that site. If the user clicks yes then it is redirected to the given input website. Else it is redirected to the index page.

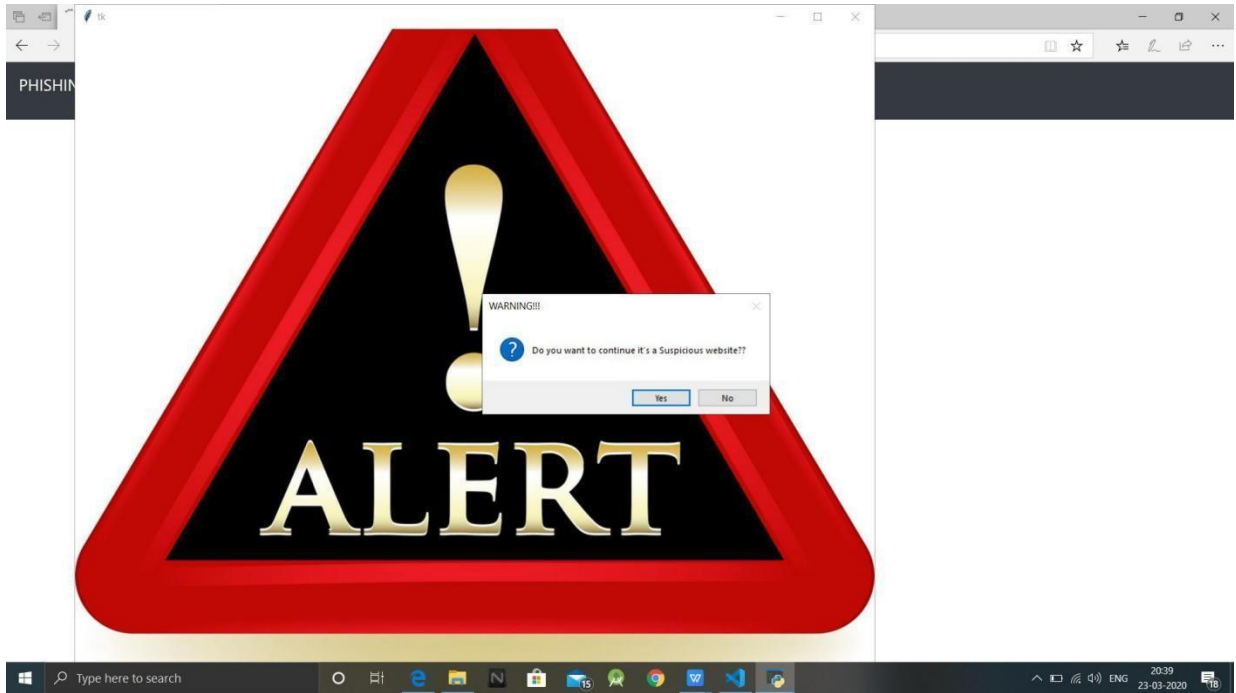


Fig 5.1.5 Suspicious website Output

This is the output page displayed if the given input url is a suspicious one. It also asks the user to redirect to that site. If the user clicks yes then it is redirected to the given input website. Else it is redirected to the index page.

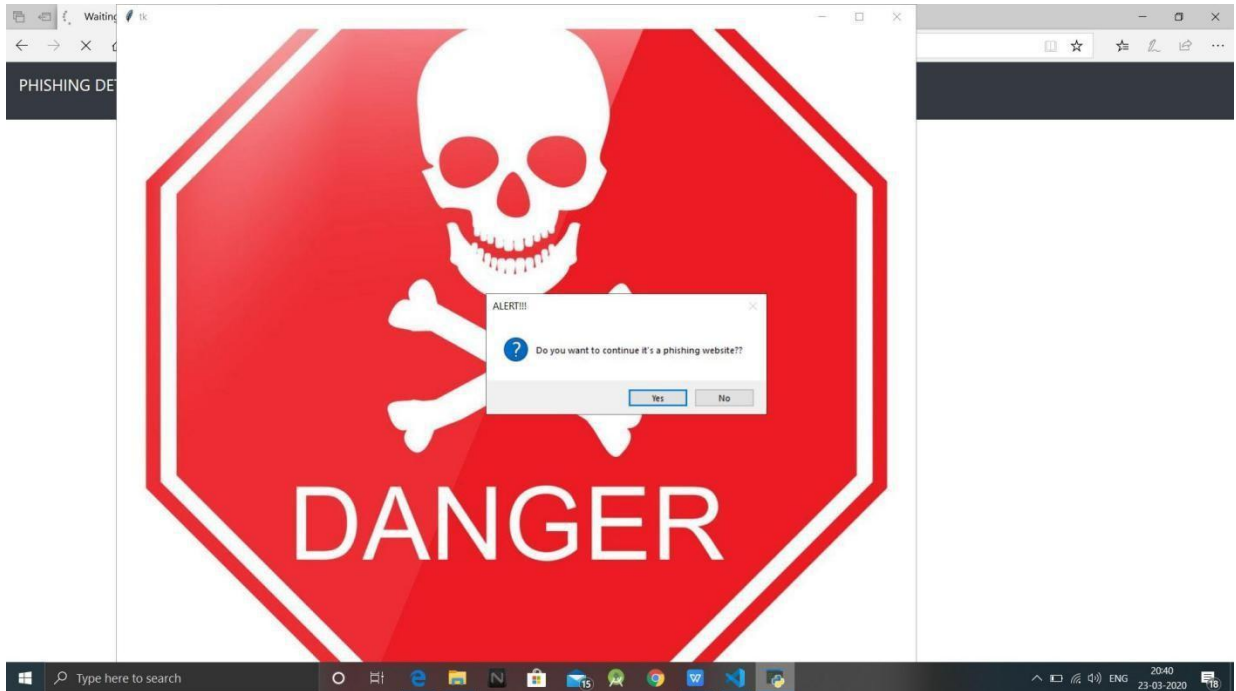


Fig 5.1.6 Phishing website Output

This is the output page displayed if the given input url is a phishing one. It also asks the user to redirect to that site. If the user clicks yes then it is redirected to the given input website. Else it is redirected to the index page.

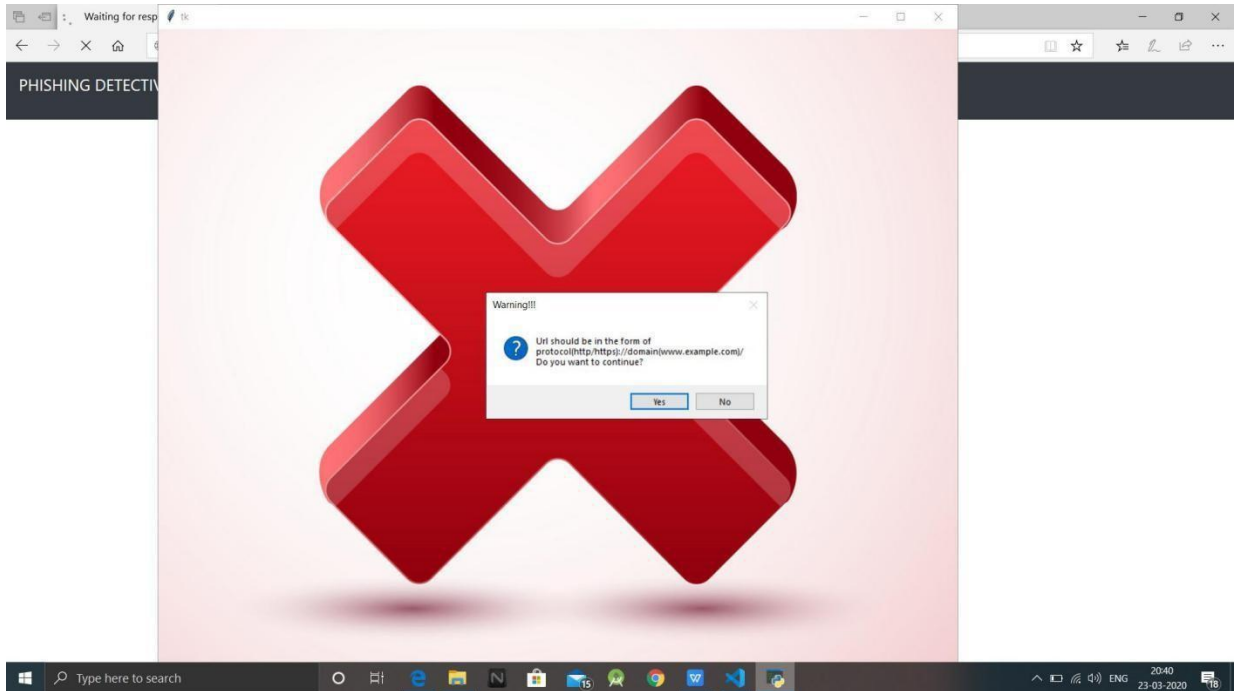


Fig 5.1.7 URL Format Error

This page displays error if the given input URL format is not in the form of “(http/https)://domain(www.example.com)”. It also asks the user to redirect to the index page. If the user clicks yes then it is redirected to the index page. Else it is redirected to login page.

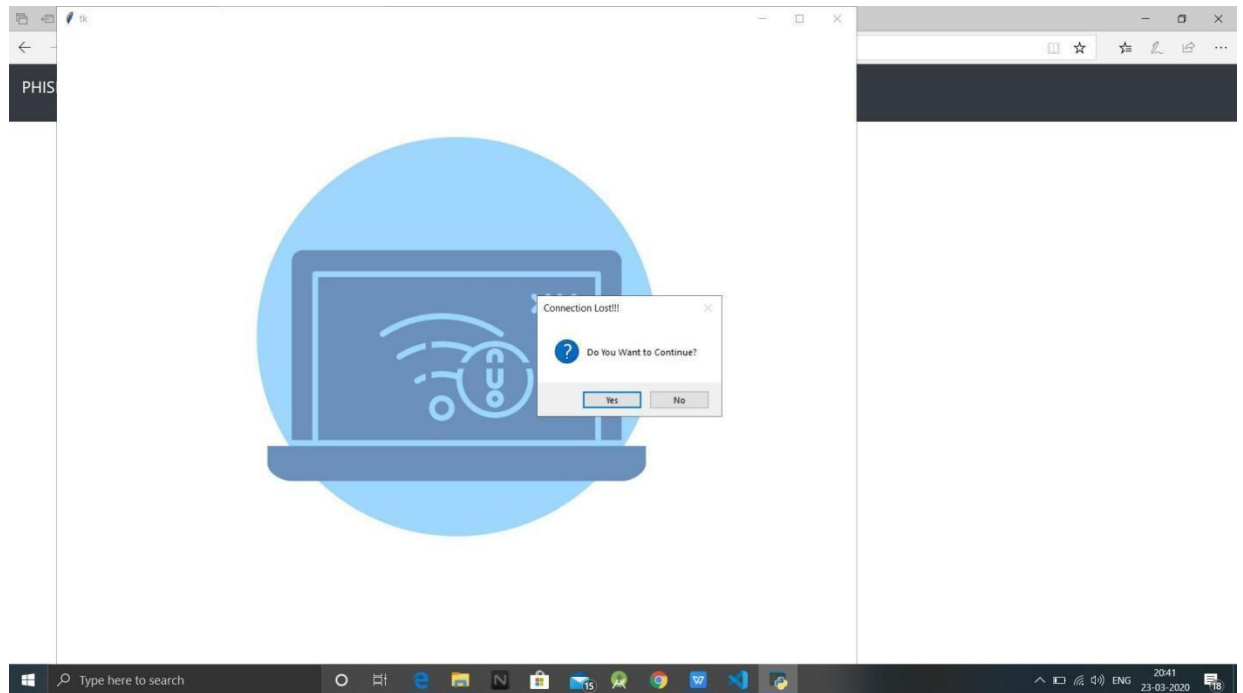


Fig 5.1.8 Connection Lost Error

This page displays error if the connection is lost. This error occurs because of the connection failure of opening of urls. It also asks the user to redirect to the index page. If the user clicks yes then it is redirected to the index page. Else it is redirected to login page.

7.2 CODING

DETECT PHISHING.py:

```
import Condition1
import Condition2
import Condition3
import Condition4
import pandas as
pd import os
import
webbrowser from
```

```

tkinter import *
import tkinter as tk
import PIL from
PIL import
ImageTk, Image
from tkinter import
messagebox
website =
os.sys.argv[1]
Condition1.catego
ry1(website)
Condition2.category2(website)
Condition3.category3(website)
Condition4.category4(website) read =
pd.read_csv(r'phishing5.txt',header = None,sep
= ',') read = read.iloc[:, :-1].values
dataset = pd.read_csv(r'Training
Dataset1.csv') X =
dataset.iloc[:, :-1].values y =
dataset.iloc[:, :-1].values
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2,random_state =
1001) from sklearn.ensemble import RandomForestRegressor regressor =
RandomForestRegressor(n_estimators = 10,criterion = "mse",random_state = 2)
regressor.fit(X_train,y_train) y_pred = regressor.predict(X_test) from
sklearn.model_selection import cross_val_score accuracy =
cross_val_score(estimator = regressor,X=X_train,y=y_train,cv = 5) accuracy.mean()
accuracy.std()
Detect_phishing_website =
regressor.predict(read) if
Detect_phishing_website == 1:
    top = tk.Tk() img = ImageTk.PhotoImage(Image.open("safe.jpg")) background =
    tk.Label(top, image=img, bd=0) background.pack(fill='both', expand=False)
    background.image = img background.rowconfigure(0, weight=100)

```

```

background.rowconfigure(3, weight=100) background.columnconfigure(0,
weight=100) background.columnconfigure(3, weight=100) reply =
messagebox.askyesno("REQUEST!!!", "Can I redirect to this Legitimate Website?") if
reply == True:
    webbrowser.op
en(website) else:
    exit()
top.mainloop() elif Detect_phishing_website > 0 and
Detect_phishing_website < 0.6:
top = tk.Tk() img =
ImageTk.PhotoImage(Image.open("alert.
jpg")) background = tk.Label(top,
image=img, bd=0)
background.pack(fill='both',
expand=False) background.image = img
background.rowconfigure(0, weight=100)
background.rowconfigure(3, weight=100)
background.columnconfigure(0,
weight=100)
background.columnconfigure(3,
weight=100) reply =
messagebox.askyesno("WARNING!!!", "D
o you want to continue it's a Suspicious
website??")
if reply == True:
    webbrowser.op
en(website) else:
    exit()
top.mainloop()
else:
top = tk.Tk() img = ImageTk.PhotoImage(Image.open("danger.jpg")) background =
tk.Label(top, image=img, bd=0) background.pack(fill='both', expand=False)
background.image = img background.rowconfigure(0, weight=100)
background.rowconfigure(3, weight=100) background.columnconfigure(0, weight=100)

```

```
background.columnconfigure(3, weight=100) reply =  
messagebox.askyesno("ALERT!!!", "Do you want to continue it's a phishing website??") if  
reply == True:  
    webbrowser.op  
en(website) else:  
    exit()  
top.mainloop()
```